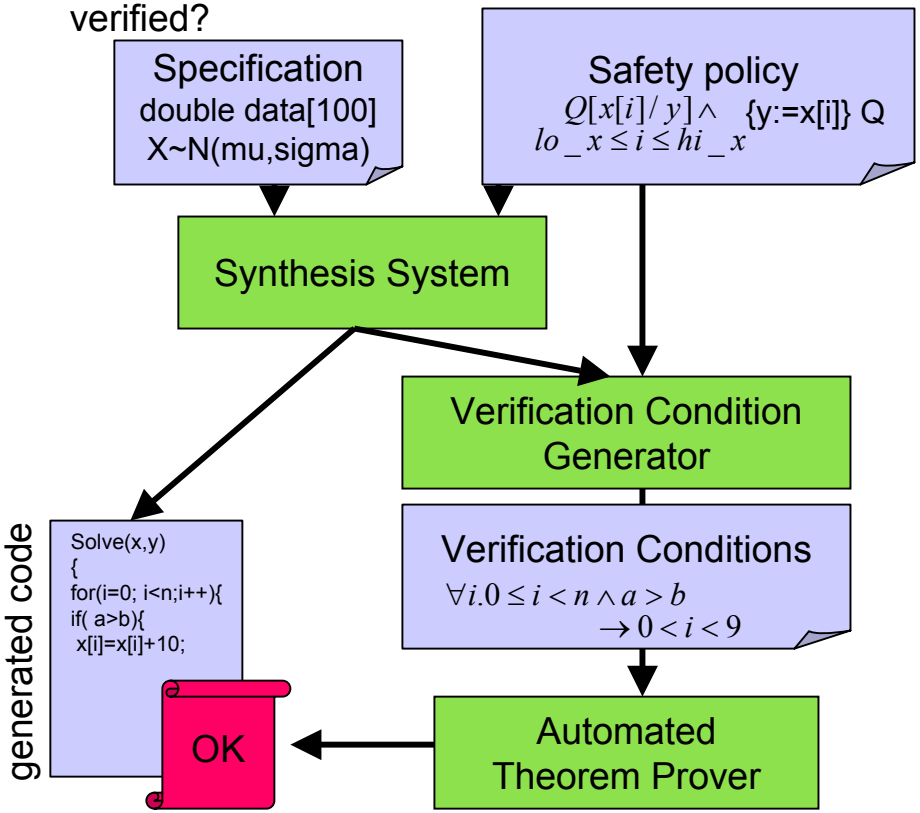
Extending Code Generators with Certification Capability

PROBLEM

Fully automatic program synthesis offers many gains over traditional software development methods. e.g., speed of development, increased adaptability and reliability. But code generators are complex pieces of software themselves that may contain bugs.

- Can you trust the code-generator?
- How can the correctness of the generated code be verified?



SOLUTION

Extend the code generator to support the verification process. Many software errors are violations of safety properties, which can be automatically verified on a program by program basis. e.g.,

- Array-bound safety,
- Variable initialization-before-use,
- Variable write limits for volatile memory, …

TECHNOLOGY

We augmented the AutoBayes/AutoFilter program synthesis systems to *automatically* generate and process code annotations for specified safety policies. A Verification Condition Generator produces simple logical conditions which are then automatically proven using an automated theorem prover. Thus, no user interaction is necessary beyond giving the initial specification.

Explanation of Accomplishment

- **POC:** Ewen Denney, Bernd Fischer (ASE Group, Code IC, edenney@email.arc.nasa.gov)
- **Milestone:** Develop program synthesis technology that enables automatic product-oriented certification, rather than certification for flight based on traditional methods.
- **Accomplishment:** The ASE group is developing synthesis systems which are able to automatically generate a wide range of complex programs in the NASA-relevant domains of data analysis and state estimation. We have extended these systems with the capability of generating annotated code which can then be automatically verified for compliance with a given safety property. The main increment over previous work is in extending the program synthesis system so that it is customizable with respect to different notions of safety. The certification is significantly more accurate (fewer false positives) than commercial analysis tools.
- **Shown:** The synthesis system takes as input a high-level specification of a program. We have extended the system to also take an explicit safety policy as input. The system then automatically generates an imperative program which meets the specification annotated with information appropriate to the safety policy. The annotated program is then passed to a verification condition generator (VCG), which uses the safety policy to generate a list of verification conditions, which can then be checked automatically by a theorem prover.
- **Benefits:** This technology has the potential to increase confidence in the use of code generators within and outside NASA. Auto-generated code will come with a certificate of its correctness, with respect to user-defined notions of safety. These certificates can be independently checked by third parties such as a certification authority.